WRDC-TR-90-8007
Volume VIII
Part 33

WRIGHT RESEARCH & DEVELOPMENT CENTER

1

INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume VIII - User Interface Subsystem
Part 33 - Application Interface Development Specification

S. Barker, F. Glandorf

Control Data Corporation
Integration Technology Services
2970 Presidential Drive
Fairborn, OH 45324-6209

September 1990

Final Report for Period 1 April 1987 - 31 December 1990

Approved for Public Release; Distribution is Unlimited

MANUFACTURING TECHNOLOGY DIRECTORATE
WRIGHT RESEARCH AND DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, regardless whether or not the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data. It should not, therefore, be construed or implied by any person, persons, or organization that the Government is licensing or conveying any rights or permission to manufacture, use, or market any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations

David Judson

DAVID L. JUDSON, Project Manager
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

FOR THE COMMANDER:

Bruce Rasmussen

BRUCE A. RASMUSSEN, Chief
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

If your address has changed, if you wish to be removed form our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/MTI, Wright-Patterson Air Force Base, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS
None

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for Public Release;
Distribution is Unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
DS 620344700

5. MONITORING ORGANIZATION REPORT NUMBER(S)
WRDC-TR-90-8007 Vol. VIII, Part 33

6a. NAME OF PERFORMING ORGANIZATION
Control Data Corporation;
Integration Technology Services

6b. OFFICE SYMBOL *(if applicable)*

7a. NAME OF MONITORING ORGANIZATION
WRDC/MTI

6c. ADDRESS *(City,State, and ZIP Code)*
2970 Presidential Drive
Fairborn, OH 45324-6209

7b. ADDRESS *(City, State, and ZIP Code)*
WPAFB, OH 45433-6533

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
Wright Research and Development Center,
ir Force Systems Command, USAF

8b. OFFICE SYMBOL *(if applicable)*
WRDC/MTI

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUM.
F33600-87-C-0464

8c. ADDRESS *(City, State, and ZIP Code)*
Wright-Patterson AFB, Ohio 45433-6533

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
78011F	595600	F95600	20950607

1. TITLE *(Include Securitv Classification)*
See block 19

2. PERSONAL AUTHOR(S)
Structural Dynamics Research Corporation: Barker, S., Glandorf, F., et al.

3a. TYPE OF REPORT
Final Report

13b. TIME COVERED
4/1/87-12/30/90

14. DATE OF REPORT *(Yr.,Mo.,Day)*
1990 September 30

15. PAGE COUNT
18

6. SUPPLEMENTARY NOTATION
WRDC/MTI Project Priority 6203

7. COSATI CODES

FIELD	GROUP	SUB GR.
1308	0905	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify block no.)*

9. ABSTRACT *(Continue on reverse if necessary and identify block number)*

This specification establishes the development, test, and qualilfication requirements of a computer program identified as the Application Interface, referred to as AI.

BLOCK 11:

INTEGRATED INFORMATION SUPPORT SYSTEM
Vol VIII -User Interface Subsystem

Part 33 - Application Interface Development Specification

0. DISTRIBUTION/AVAILABILITY OF ABSTRACT
NCLASSIFIED/UNLIMITED x SAME AS RPT. DTIC USERS

1. ABSTRACT SECURITY CLASSIFICATION
Unclassified

2a. NAME OF RESPONSIBLE INDIVIDUAL
David L. Judson

2b. TELEPHONE NO. *(Include Area Code)*
(513) 255-7371

2c. OFFICE SYMBOL
WRDC/MTI

EDITION OF 1 JAN 73 IS OBSOLETE

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

FOREWORD

This technical report covers work performed under Air Force Contract F33600-87-C-0464, DAPro Project. This contract is sponsored by the Manufacturing Technology Directorate, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Bruce A. Rasmussen, Branch Chief, Integration Technology Division, Manufacturing Technology Directorate, through Mr. David L. Judson, Project Manager. The Prime Contractor was Integration Technology Services, Software Programs Division, of the Control Data Corporation, Dayton, Ohio, under the direction of Mr. W. A. Osborne. The DAPro Project Manager for Control Data Corporation was Mr. Jimmy P. Maxwell.

The DAPro project was created to continue the development, test, and demonstration of the Integrated Information Support System (IISS). The IISS technology work comprises enhancements to IISS software and the establishment and operation of IISS test bed hardware and communications for developers and users.

The following list names the Control Data Corporation subcontractors and their contributing activities:

SUBCONTRACTOR	ROLE
Control Data Corporation	Responsible for the overall Common Data Model design development and implementation, IISS integration and test, and technology transfer of IISS.
D. Appleton Company	Responsible for providing software information services for the Common Data Model and IDEF1X integration methodology.
ONTEK	Responsible for defining and testing a representative integrated system base in Artificial Intelligence techniques to establish fitness for use.
Simpact Corporation	Responsible for Communication development.
Structural Dynamics Research Corporation	Responsible for User Interfaces, Virtual Terminal Interface,and Network Transaction Manager design, development, implementation, and support.
Arizona State University	Responsible for test bed operations and support.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

Figure | Page

SECTION 1

SCOPE

1.1 Identification

This specification establishes the development, test and qualification requirements of a computer program identified as the Application Interface, referred to as the AI. The AI is one configuration item of the Integrated Information Support System (IISS) User Interface (UI).

1.2 Functional Summary

The AI is a collection of procedures that may be linked with an application to enable it to use the Form Processor (FP) and run in the distributed IISS environment. The AI does this by sending/receiving FP requests through the NTM (Network Transaction Manager) to/from the User Interface Monitor (UIM) of the Form Processor.

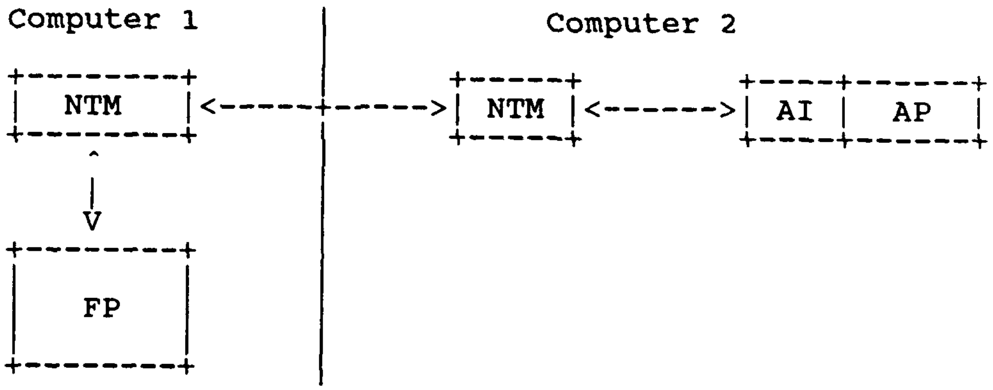

Figure 1-1 Distributed IISS Environment

SECTION 2

DOCUMENTS

2.1 Reference Documents

[1] Control Data Corporation, System Design Specification, 31 May 1988.

[2] Structural Dynamics Research Corporation, Form Processor Development Specification, DS 620344200, 31 May 1988.

[3] Structural Dynamics Research Corporation, Forms Language Compiler Development Specification, DS 620344401, 31 May 1988.

[4] Structural Dynamics Research Corporation, Forms Driven Form Editor Development Specification, DS 620344402, 31 May 1988.

[5] Structural Dynamics Research Corporation, Report Writer Development Specification, DS 620344700, 31 May 1988.

[6] Structural Dynamics Research Corporation, Rapid Application Generator Development Specification, DS 620344502, 31 May 1988.

[7] Structural Dynamics Research Corporation, Text Editor Development Specification, DS 620344600, 31 May 1988.

[8] Structural Dynamics Research Corporation, User Interface Services Development Specification, DS 620344100, 31 May 1988.

[9] Structural Dynamics Research Corporation, Virtual Terminal Interface Development Specification, DS 620344300, 31 May 1988.

[10] Systran, ICAM Documentation Standards, 15 September 1983.

[11] Structural Dynamics Research Corporation, Form Processor User Manual, UM 620344200, 31 May 1988.

2.2 Terms and Abbreviations

American Standard Code for Information Interchange: (ASCII), the character set defined by ANSI X3.4 and used by most computer vendors.

Application Definition Language: an extension of the Forms Definition Language that includes retrieval of database information and conditional actions. It is used to define interactive application programs.

Application Generator: (AG), subset of the IISS User Interface that consists of software modules that generate IISS application code and associated form definitions based on a language input. The part of the AG that generates report programs is called the Report Writer. The part of the AG that generates interactive applications is called the Rapid Application Generator.

Attribute: field characteristic such as blinking, highlighted, black, etc. and various other combinations. Background attributes are defined for forms or windows only. Foreground attributes are defined for items. Attributes may be permanent, i.e., they remain the same unless changed by the application program, or they may be temporary, i.e., they remain in effect until the window is redisplayed.

Buffer Name: the default file in which the buffer will be saved if no file is given on a save command.

Common Data Model: (CDM), IISS subsystem that describes common data application process formats, form definitions, etc. of the IISS and includes conceptual schema, external schemas, internal schemas, and schema transformation operators.

Conceptual Schema: (CS), the standard definition used for all data in the CDM. It is based on IDEF1 information modeling.

Current Cursor Position: the position of the cursor before an edit command or function is issued in the text editor.

Cursor Position: the position of the cursor after any command is issued.

Cut and Paste Buffer: where deleted lines go and the paste and fill edit commands get their data.

Device Drivers: (DD), software modules written to handle I/O for a specific kind of terminal. The modules map terminal specific commands and data to a neutral format. Device Drivers are part of the UI Virtual Terminal.

Display List: a list of all the open forms that are currently being processed by the Form Processor or the user.

Display Start Line: the first line in the buffer to be displayed.

Display Size: the number of lines used in the edit area.

Extended Binary Coded Decimal Interchange Code: (EBCDIC), the character set used by a few computer vendors (notably IBM) instead of ASCII.

External Schema: (ES), an application's view of the CDM's conceptual schema.

Field Pointer: indicates the ITEM which contains the current cursor position.

Forms Driven Form Editor: (FDFE), subset of the FE which consists of a forms driven application used to create Form Definition files interactively.

Form Editor: (FE), subset of the IISS User Interface that is used to create definitions of forms. The FE consists of the Forms Driven Form Editor and the Forms Language Compiler.

Forms Language Compiler: (FLAN), subset of the FE that consists of a batch process that accepts a series of forms definition language statements and produces form definition files as output.

Form Processor Text Editor: (FPTE), subset of the Form Processor that consists of software modules that provide text editing capabilities to all users of applications that use the Form Processor.

Integrated Information Support System: (IISS), a computing environment used to investigate, demonstrate, test the concepts and produce application for information management and information integration in the context of Aerospace Manufacturing. The IISS addresses the problems of integration of data resident on heterogeneous data bases supported by heterogeneous computers interconnected via a Local Area Network.

Item: non-decomposable area of a form in which hard-coded descriptive text may be placed and the only defined areas where user data may be input/output.

Logical Device: a conceptual device that identifies a top level window of an application. It is used to distinguish between multiple applications running simultaneously on a physical device. NOTE: a single application can have more than one logical device. To the end user this also appears as multiple applications running simultaneously.

Neutral Data Manipulation Language: (NDML), the command language by which the CDM is accessed for the purpose of extracting, deleting, adding, or modifying data.

Open List: a list of all the forms that are currently open for an application process.

Operating System: (OS), software supplied with a computer which allows it to supervise its own operations and manage access to hardware facilities such as memory and peripherals.

Page: instance of forms in windows that are created whenever a form is added to a window.

Paging and Scrolling: a method which allows a form to contain more data than can be displayed with provisions for viewing any portion of the data buffer.

Physical Device: a hardware terminal.

Presentation Schema: (PS), may be equivalent to a form. It is the view presented to the user of the application.

Previous Cursor Position: the position of the cursor when the previous edit command was issued.

Previous Edit Command: the function key pressed before the current one.

Rapid Application Generator: (RAP), part of the Application Generator that generates source code for interactive programs based on a language input.

Report Definition Language: an extension of the Forms Definition Language that includes retrieval and calculation of database information and is used to define reports.

Report Writer: (RW), part of the Application Generator that generates source code for report programs based on a language input.

Select Line: one terminus of the select range.

Select Mode: when on, certain commands will be executed over the lines in the selected range. The commands are <DELETELINE> and replace.

Subform: a form that is used within another form.

Text Editor: (TE), subset of the IISS User Interface tnat consists of a file editor that is based on the text editing functions built into the Form Processor.

Top of file: the first line of the buffer.

User Interface Development System: (UIDS), collection of IISS User Interface subsystems that are used by applications programmers as they develop IISS applications. The UIDS includes the Form Editor and the Application Generator.

User Interface Monitor: (UIM), part of the Form Processor that handles messaging between the NTM and the UI. It also provides authorization checks and initiates applications.

User Interface Services: (UIS), subset of the IISS User Interface that consists of a package of routines that aid users in controlling their environment. It includes message management, change password, and application definition services.

Virtual Terminal Interface: (VTI), the callable interface to the VT.

Virtual Terminal Interface Field Map: defines the complete terminal screen by breaking it into pieces of the various forms and items that are displayed. Each area of the terminal screen must be defined as belonging to a particular field in the display list.

Window Manager: a facility which allows the following to be manipulated: size and location of windows, the device on which an application is running, the position of a form within a window. It is part of the Form Processor.

SECTION 3

REQUIREMENTS

This section includes functional and performance requirements for the AI. In addition, the AI interfaces to other IISS Computer Program Configuration Item's (CPCI's) are defined.

3.1 Computer Program Definition

Application programs use the FP Application Interface (AI) to create the messages which correspond to FP calls and are sent to the User Interface Monitor (UIM) of the Form Processor by way of the Network Transaction Manager (NTM). The UIM then decodes these messages and calls the appropriate Form Processor routine. This enables applications to run under IISS in a distributed environment.

3.1.1 System Capacities

No capacity requirements have been identified for the AI.

3.1.2 Interface Requirements

The AI interfaces with application programs and the UIM. The UIM receives the message formatted by the AI and translates it into a call to the appropriate FP routine. The Form Processor routines then interface with the VT by translating an application request into the appropriate VT command when input/output is necessary.

The following describes a typical application/AI transaction:

1) The application calls the AI procedure.

2) The AI procedure creates a message using the FP procedure's id number and input parameters and sends the message.

3) The UIM receives the message and calls the corresponding FP procedure.

4) The UIM creates a message using the FP procedure's output parameters and sends the message.

5) The AI procedure receives the message and returns the contents to the application via the argument list.

3.1.2.1 Interface Block Diagram

The structure of the AI interfaces is shown in Figure 3-1.

```
+-------------------+
|    application    |
|                   |
+-------------------+
|                   |
|        AI         |
+---------+---------+
          ^
          |  messages
          |
          V
+---------+---------+
|                   |
|        NTM        |
+---------+---------+
          ^
          |  messages
          |
          V
+---------+---------+
|                   |
|      UIM/FP       |
+-------------------+
```

Figure 3-1 AI Interfaces

3.1.2.2 Detailed Interface Definition

3.1.2.2.1 Application

The interface to the application is identical to that of the Form Processor procedures and is documented in the FP User Manual.

3.1.2.2.2 User Interface Monitor

The AI sends and receives messages to and from the UIM via the NTM. The format of these messages is specified in Section 3.4.

3.2 Detailed Functional Requirements

The AI enables an application to run under IISS in a distributed environment. It provides an application with a set of procedures whose names, number and types of arguments and functionality are identical to that of the Form Processor. These procedures are documented in the Form Processor Development Specification.

3.3 Special Requirements

3.3.1 Programming Methods

The AI is programmed in C using structured design with care taken to insure portability of the the AI code with minimum effort. Basic programming standards for readability and ease of debugging shall be followed.

3.3.2 Expandability

Since the AI has been designed as a set of interface routines, new functionality may be added by simply adding new interface routines.

3.4 NTM Message Format

Messages from the AI to the UIM contain the following fields:

- o An integer representing the AI version number,
- o An integer representing the routine to be called,
- o The integer input parameters of the routine, and
- o The integral length of implied length string input parameters (such as path names), the fixed length string input parameters, and the variable length string input parameters.

Messages from the UIM to the AI contain the following fields:

- The return code parameter from the routine,
- The integer output parameters from the routine, and
- The integral length of implied length string output parameters, the fixed length string output parameters (other than the return code), and the variable length string output parameters.

In messages, integers are represented in character format as a variable length string of digits with an optional preceding minus sign (such as is generated by a "%d" format in C) followed by a space terminator. Strings are copied directly into the message without any terminator.

SECTION 4

QUALITY ASSURANCE PROVISIONS

4.1 Introduction and Definitions

The AI is tested using the following tests:

Computer programming test and evaluation. This testing primarily involves testing all the AI interface routines and internal functions for correct processing and output.

System test. This testing involves testing all the AI interface routines and internal functions within the integrated system.

4.2 Computer Programming Test and Evaluation

The test developed for the AI consists of another computer program which simulates the use of the AI interfaces routines by an application program. Every AI interface routine is exercised directly by the computer test program and every internal function is indirectly exercised by the computer test program. Variations on the computer test program can be provided by user interaction with the computer test program.

The same computer test program is run to test the AI during the system test process with the other components of the IISS.

SECTION 5

PREPARATION FOR DELIVERY

The implementation site for the constructed software is the Integrated Support System (IISS) Test Bed site located at Arizona State University at Tempe, Arizona. The software associated with each AI CPCI release is delivered on a media which is compatible with the IISS Test Bed. The release is clearly identified and will include instructions on procedures to be followed for installation of the release.

SECTION 6

NOTES

Please refer to the Software Availability Bulletin, Volume III, Part 16, CI# SAB620326000, for current IISS software and documentation availability.

U.S. Government Printing Office 648-127